「効率的遡及入力方法」について - 主題別研究集会の講演要旨 -

AUTHOR(S):

宮沢, 彰

CITATION:

宮沢, 彰. 「効率的遡及入力方法」について - 主題別研究集会の講演要旨 -. 静脩 1990, 27(1): 7-8

ISSUE DATE:

1990-07

URL:

http://hdl.handle.net/2433/37075

「効率的遡及入力方法」について
―主題別研究集会の講演要旨―

学術情報センター教授　宮　沢　　彰

最初におことわりしなければならないのですが、私のこの研究はまだ完成段階に入ったものではありません。とは言っても先日出ました科研費の報告書「大量文献情報遡及変換入力システムの高度化に関する研究」の中に中間報告を書いております。今日のお話はこの報告から進んだものではありません。同じ報告書の中により完成した研究である富山大学の米田・長谷先生の報告、名古屋大学の渡辺先生他の報告、九州大学の松尾先生他の報告等ありますのでそれらも参照していただければと思います。

もう１つ、この研究ではルールベースのプログラミングといういわゆるＡＩ的手法を用いています。しかし目的はＡＩ的手法の開発にあるのではなく、データベース作成という応用の中で、こういった手法がどこまで有効であるかテストすることにあります。このためＡＩ的プログラミングそのものに興味をお持ちの方には期待はずれかもしれませんが、お許しください。

本題に入りましょう。目録の遡及入力というのは、カード等のこれまでの目録を、コンピュータで扱えるレコードにして（遡及変換）、総合目録等データベースに入力する（遡及入力）作業です。目録作成の機械化が進展して来た現在、図書館の目録全体をデータベースとするために必須の作業となっています。この作業のこれまでの方法にはオンライン入力とバッチ入力、その折衷法等ありますが、いずれも人手・コストのかかる方法です。

オンライン入力は新しい本の整理の時と同様にカードを見ながら端末に向って入力して行きます。単純な方法ですがヒット率が高い時には最も安価で効率的な方法です。

バッチ入力のこれまでの方法は、カード上の標目、標題、出版事項等をマークする「タグ付け」、機械可読にする「パンチ入力」、既存のデータベースとの重複を取除く等整合させる「照合調整」といった工程で行なうものでした。

これらの工程は殆ど人手によるものでしたが、今回の研究はこのうちの「タグ付け」を機械で自動的に行なおうというものです。もちろん「パンチ入力」の部分の機械化であるＯＣＲ入力と組み合わせればより一層の機械化が達成できますが、（これも考えてはいますが）今回の話とは別にしておきます。

目録カードを入力したものは、配置を持った文字列になります（図１）。ここからどの部分がどういうフィールドかを認識してレコードを作る（図２）のがフィールドの自動認識です。目録カードの文字列が文法に従っていてそれを解析するというのであればよく知られた技術ですが、残念ながら目録にはそういった文法はありません。

そこで使うのが、㎝がついていれば大きさだろうとか、1984のような形であれば出版年であろうとか、１行目から始まっていれば標目（多くは著者名）だろう、といったような「知識」をもとに残りを推定して行く方法です。一見危なそうで、事実なかなか100％の認識はむづかしいのですが、十分な「知識」をもてば実用的な認識は可能になるだろうという予想なわけです。

実際に使用したシステムは、ＳＵＮ３というワークステーションとＯＰＳ83というルールベースの言語です。文字列のパターンマッチにはレギュラーエクスプレッションを使用する方法をとりました。ＯＰＳ83を使用した理由は、たまたま手元にあったから、というのもありますが、1)Ｃ言語等で書かれた外部ルーチンを利用できる、2)手続き型言語とルールベース言語の双方を持っていて、実際的なシステムが組み易い、3)実行速度が速い（と言われる）などのためです。

この手続き型とルールベースというのがわかり

にくいかと思います。普通のプログラミング言語FORTRANとかCとかBASICとかはみな手続き型言語で、こうしてああしてこうだったら次にこうして…、というようにプログラムします。これに対し、ルールベースというのは、こうだったらこうする、というルールを、処理の順番とは関係なく並べてプログラムします。そんなのでうまくいくのか、という疑問を持たれると思いますが、うまくいく様にプログラムすればうまく行きます。残念ながら、これによってプログラムを書くのが易しくなったとか、わかり易くなったとかいうものではありません。

ルールベースの話をもう少し。使ってみてこの方式の良い点は、ルールの追加や変更に伴う副作用が少く、メンテナンスしやすいことです。最初は最小限のルールで開始して、うまくいかないデータを見つける度に、分析してルールを追加・変更するという開発方法がとれます。手続き型では処理の前の方を変更すると後の方に影響が出ることがしばしばですが、そのような心配があまりありません。

ただし、カード1枚分を読んで、パラメータの指定によりプリントしたり、行数、枚数を数えたりして、という仕事は手続き型で書く方がはるかに自然です。また、1×××という4桁の数字が最後にあって、という文字列のパターンマッチもアルゴリズムが色々開発されていますので、手続き型で書くのが楽です。今回はレギュラーエクスプレッションによるパターンマッチのライブラリ関数を用いました。このようにC言語用のライブラリを容易に使える点もＯＰＳ83を用いた理由の1つです。

さて、このような形でシステムを作りまして、東大の全学総合目録カードの中から無作為に選んだ72件を処理した結果、正しく認識したのが44件約61％でした。まだ中間的な結果なのですが、実用レベルには少し遠いようです。ルールが対応しきれていないという理由と、形態的特徴による認識の限界という理由によるものと思います。

ルールが対応しきれていない理由の1つは、この72枚の中にさえ、英語、ドイツ語、フランス語、イタリア語、中国語、日本語（ローマ字）と多くの言語が出現していることです。これはルールを増やしていくしかないでしょう。もう1つの形態的特徴というのは、区切字やいくつかのキーワードによる認識（cmなど）です。形態的特徴によって例えば「××、1896」というのが出版事項と認識しても、「××」が出版地か出版者かは地名や人名を知らないと判断できません。これに対してはデータベース参照による判断を組み込む必要があるのではないかと考えています。

最後に、感想を含めて今後の見通しを。結論から言いますと、目録のわかった人とルールの書ける人2～3人でかなり集中して時間を使えば、実用レベルには達するのではないかと思います。と言っても、どこまでやれば終りと言えるのかわからない、という悩みがあります。極端な事を言えば、東大のカードを全部入れて確認した時にやっと（東大用として）完成した事になる。実際にはそこまで行かなくとも、元の目録の誤りと同程度の誤認識率ならばほゞ実用的ではありましょう。それにしても、どこまでやればそうなるかについてはもう少し経験をつむ必要がありそうです。

もう1つは、このアプローチをオンライン入力方式と比較した時の問題点です。オンライン方式は、レコードがヒットすれば記述を入力する必要がありません。これに対しこの方式ではともかくも全文字を一旦は入力しなければならないわけで、ヒット率の高い場合のオンライン入力より効率的にするにはＯＣＲ利用が不可欠でしょう。

```
Chibret, [ Paul ].
Astigmatisme selon et contre la regle resultats compares de l'examen
objectif (keratometrie, skiascopie) et de l'examen subjectif.
    Par., Steinheil, 1890 . 8o (Pm . )
(Bd.w. : Exploratio oculi, Untersuchung des Aiges ( 4 ). Refractio
et accommodatio oculi.
Refraction und Accommodation ( 8 ). Astigmatismus. Abhandlungen:
347-9. )
```

図－1　カード入力例

```
HDNG : Chibret, [ Paul ].
TR : Astigmatisme selon et contre la regle resultats compares de
     l'examen objectif (keratometrie, skiascopie) et de l'examen
     subjectif
PUB.P : Par
PUB.N : Steinheil
PUB.Y : 1890
PHYS.S : 8o (Pm . )
NOTE.PTB : (Bd.w.: Exploratio oculi, Untersuchung des Aiges ( 4 ).
           Refractio et accommodatio oculi. Refraction und
           Accommodation ( 8 ). Astigmatismus. Abhandlungen:
           347-9 . )
```

図－2　認識結果